

What is claimed is:

- 1 1: A method comprising:
 - 2 processing a group of instructions in topological dependence order;
 - 3 selecting an instruction, associated with at least one operand, to schedule;
 - 4 allocating a register to one or more of the operands;
 - 5 determining if the live range of the operand spans a function call;
 - 6 if so, attempting to allocate a preserved register;
 - 7 if not, attempting to allocate a scratch register; and
 - 8 if the determination is unknown, treating the live range as tentative; and
 - 9 if the allocation was successful, scheduling the instruction.

- 1 2: The method of claim 1, wherein allocating a register to one or more of the operand
 - 2 includes:
 - 3 determining if the operand's use begins a live range or ends a live range;
 - 4 if the operand's use begins a live range, attempting to allocate a register;
 - 5 if the operand's use ends a live range, marking the register associated with the
 - 6 operand as available for reallocation.

- 1 3: The method of claim 1, wherein attempting to allocate a preserved register includes:
 - 2 determining if a preserved register is available for allocation from a list of
 - 3 previously used preserved registers; and
 - 4 if so, allocating the available preserved register for the live range defined by the operand.

1 4: The method of claim 3, further including, if a preserved register is not available from
 2 the list of previously used preserved registers,
 3 attempting to find a tentative live range that includes a preserved register that is in
 4 the list of previously used preserved registers; and
 5 if successful, allocating this preserved register for the live range defined by the operand,
 6 and reserving a different preserved register for the tentative live range.

1 5: The method of claim 4, further including, if the attempt to find a tentative live range
 2 that includes a preserved register that is in the list is not successful,
 3 adding a new preserved register to the list of previously used preserved registers; and
 4 allocating this new preserved register to the live range defined by the operand.

1 6: The method of claim 1, wherein attempting to allocate a scratch register includes:
 2 determining if a scratch register is available for allocation;
 3 if so, allocating a scratch register for the live range defined by the operand;
 4 if not,
 5 attempting to steal a scratch register from a live range marked as tentative;
 6 determining if the attempted theft was successful;
 7 if so, allocating the stolen scratch register to the live range defined by the

8 operand; and
9 if not, attempting to allocate a preserved register.

1 7: The method of claim 1, wherein treating the live range as tentative further includes:
2 if both a scratch register and a preserved register are available for allocation,
3 reserving both the scratch register and the preserved register to the live range;
4 allowing another instruction to be processed; and
5 waiting for more information about the live range.

1 8: The method of claim 7, further including, when more information is available:
2 if the last use of the live range has been scheduled, allocating the reserved scratch
3 register to the live range defined by the operand, and freeing the reserved preserved
4 register;
5 if a function call has been scheduled before the last use of the live range or the
6 reserved scratch register has been stolen, allocating the preserved register to the live
7 range defined by the operand and freeing the reserved scratch register.

1 9: The method of claim 7, wherein reserving a preserved register for the tentative live
2 range includes:
3 preferring to reserve a currently unused preserved register that is in the list of

4 previously used preserved registers, before reserving a new unused preserved register for
5 the tentative live range.

1 10: The method of claim 9, wherein treating the live range as tentative further includes:
2 if a scratch register is not available for allocation, attempting to allocate a
3 preserved register to the live range.

1 11: The method of claim 1, wherein allocating a register to at least one or more of
2 operands associated with the instruction includes:
3 if no proper registers are available for allocation, inserting a register spill; and
4 either re-attempting to allocate a register, or selecting another instruction to
5 schedule.

1 12: An article comprising:
2 a machine readable medium having a plurality of machine accessible instructions,
3 wherein when the instructions are executed, the instructions provide for:
4 processing a group of instructions in topological dependence order;
5 selecting an instruction, associated with at least one operand, to schedule;
6 allocating a register to one or more of the operands;
7 determining if the live range of the operand spans a function call;
8 if so, attempting to allocate a preserved register;

9 if not, attempting to allocate a scratch register; and
10 if the determination is unknown, treating the live range as tentative; and
11 if the allocation was successful, scheduling the instruction.

1 13: The article of claim 12, wherein the instructions providing for allocating a register to
2 one or more of the operand includes instructions providing for:
3 determining if the operand's use begins a live range or ends a live range;
4 if the operand's use begins a live range, attempting to allocate a register;
5 if the operand's use ends a live range, marking the register associated with the
6 operand as available for reallocation.

1 14: The article of claim 12, wherein the instructions providing for attempting to allocate
2 a preserved register includes instructions providing for:
3 determining if a preserved register is available for allocation from a list of
4 previously used preserved registers; and
5 if so, allocating the available preserved register for the live range defined by the operand.

1 15: The article of claim 14, further including instructions providing for, if a preserved
2 register is not available from the list of previously used preserved registers,
3 attempting to find a tentative live range that includes a preserved register that is in
4 the list of previously used preserved registers; and

5 if successful, allocating this preserved register for the live range defined by the operand,
6 and reserving a different preserved register for the tentative live range.

1 16: The article of claim 15, further including instructions providing for, if the attempt to
2 find a tentative live range that includes a preserved register that is in the list is not
3 successful,
4 adding a new preserved register to the list of previously used preserved registers; and
5 allocating this new preserved register to the live range defined by the operand.

1 17: The article of claim 12, wherein the instructions providing for attempting to allocate
2 a scratch register includes instructions providing for:
3 determining if a scratch register is available for allocation;
4 if so, allocating a scratch register for the live range defined by the operand;
5 if not,
6 attempting to steal a scratch register from a live range marked as tentative;
7 determining if the attempted theft was successful;
8 if so, allocating the stolen scratch register to the live range defined by the
9 operand; and
10 if not, attempting to allocate a preserved register.

1 18: The article of claim 12, wherein the instructions providing for treating the live range
 2 as tentative further includes instructions providing for:
 3 if both a scratch register and a preserved register are available for allocation,
 4 reserving both the scratch register and the preserved register to the live range;
 5 allowing another instruction to be processed; and
 6 waiting for more information about the live range.

1 19: The article of claim 18, further including instructions providing for, when more
 2 information is available:
 3 if the last use of the live range has been scheduled, allocating the reserved scratch
 4 register to the live range defined by the operand, and freeing the reserved preserved
 5 register;
 6 if a function call has been scheduled before the last use of the live range or the
 7 reserved scratch register has been stolen, allocating the preserved register to the live
 8 range defined by the operand and freeing the reserved scratch register.

1 20: The article of claim 18, wherein the instructions providing for reserving a preserved
 2 register for the tentative live range includes instructions providing for:
 3 preferring to reserve a currently unused preserved register that is in the list of
 4 previously used preserved registers, before reserving a new unused preserved register for
 5 the tentative live range.

1 21: The article of claim 20, wherein the instructions providing for treating the live range
 2 as tentative further includes instructions providing for:
 3 if a scratch register is not available for allocation, attempting to allocate a
 4 preserved register to the live range.

1 22: The article of claim 12, wherein the instructions providing for allocating a register to
 2 at least one or more of operands associated with the instruction includes instructions
 3 providing for:
 4 if no proper registers are available for allocation, inserting a register spill; and
 5 either re-attempting to allocate a register, or selecting another instruction to
 6 schedule.

1 23: An apparatus comprising:
 2 an instruction scheduler to:
 3 process a group of instructions in topological dependence order, and
 4 select an instruction, associated with at least one operand, to schedule
 5 schedule an instruction, if the register allocation was successful; and
 6 a register allocator to:
 7 allocate register to a live range associated with one or more operand,
 8 determine if the live range of the operand spans a function call,
 9 if so, attempt to allocate a preserved register,

10 if not, attempt to allocate a scratch register, and
 11 if the determination is unknown, treat the live range as tentative.

1 24: The apparatus of claim 23, wherein the register allocator is capable of:
 2 determining if the operand's use begins a live range or ends a live range;
 3 if the operand's use begins a live range, attempting to allocate a register;
 4 if the operand's use ends a live range, marking the register associated with the
 5 operand as available for reallocation.

1 25: The apparatus of claim 23, wherein the register allocator is capable of, when
 2 attempting to allocate a preserved register:
 3 determining if a preserved register is available for allocation from a list of
 4 previously used preserved registers; and
 5 if so, allocating the available preserved register for the live range defined by the operand .

1 26: The apparatus of claim 25, wherein the register allocator is capable of, if a preserved
 2 register is not available from the list of previously used preserved registers:
 3 attempting to find a tentative live range that includes a preserved register that is in
 4 the list of previously used preserved registers; and
 5 if successful, allocating this preserved register for the live range defined by the operand,
 6 and reserving a different preserved register for the tentative live range.

1 27: The apparatus of claim 26, wherein the register allocator is capable of, if the attempt
2 to find a tentative live range that includes a preserved register that is in the list is not
3 successful:
4 adding a new preserved register to the list of previously used preserved registers; and
5 allocating this new preserved register to the live range defined by the operand.

1 28: The apparatus of claim 23, wherein the register allocator is capable of, when
2 attempting to allocate a scratch register:
3 determining if a scratch register is available for allocation;
4 if so, allocating a scratch register for the live range defined by the operand;
5 if not,
6 attempting to steal a scratch register from a live range marked as tentative;
7 determining if the attempted theft was successful;
8 if so, allocating the stolen scratch register to the live range defined by the
9 operand; and
10 if not, attempting to allocate a preserved register.

1 29: The apparatus of claim 23, wherein the register allocator is capable of, when treating
2 a live range as tentative:
3 if both a scratch register and a preserved register are available for allocation,

4 reserving both the scratch register and the preserved register to the live range;
 5 allowing another instruction to be processed; and
 6 waiting for more information about the live range.

1 30: The apparatus of claim 29, wherein the register allocator is capable of, when more
 2 information is available:
 3 if the last use of the live range has been scheduled, allocating the reserved scratch
 4 register to the live range defined by the operand, and freeing the reserved preserved
 5 register;
 6 if a function call has been scheduled before the last use of the live range or the
 7 reserved scratch register has been stolen, allocating the preserved register to the live
 8 range defined by the operand and freeing the reserved scratch register.

1 31: The apparatus of claim 30, wherein the register allocator is capable of, when
 2 reserving a preserved register for the tentative live range
 3 preferring to reserve a currently unused preserved register that is in the list of previously
 4 used preserved registers, before reserving a new unused preserved register for the
 5 tentative live range .

1 32: The apparatus of claim 31, wherein the register allocator is capable of, when treating
 2 a live range as tentative:

3 if a scratch register is not available for allocation, attempting to allocate a
4 preserved register to the live range.

1 33. The apparatus of claim 23, the register allocator is capable of:
2 if no proper registers are available for allocation, inserting a register spill; and
3 either re-attempting to allocate a register, or selecting another instruction to
4 schedule.

1 34. The apparatus of claim 23, wherein the register allocator includes:
2 a scratch register allocator to allocate scratch registers to live ranges that do not
3 span a function call;
4 a preserved register allocator to allocate preserved registers to live ranges that
5 span a function call; and
6 a tentative register allocator to allocate either a scratch or a preserved register to
7 live ranges that are not immediately known whether or not they span a function call.

1 35: A system comprising:
2 a set of instructions to be compiled;
3 a set of preserved registers capable of storing values that are to be preserved across
4 function calls;
5 a set of scratch registers capable to storing values that do not need to be preserved across

6 function calls; and
7 an integrated compiler having:
8 an instruction scheduler to:
9 process a group of instructions in topological dependence order, and
10 select an instruction, associated with at least one operand, to schedule
11 schedule an instruction, if the register allocation was successful; and
12 a register allocator to:
13 allocate register to a live range associated with one or more operand,
14 determine if the live range of the operand spans a function call,
15 if so, attempt to allocate a preserved register,
16 if not, attempt to allocate a scratch register, and
17 if the determination is unknown, treat the live range as tentative.

1 36: The system of claim 35, wherein the register allocator is capable of:
2 determining if the operand's use begins a live range or ends a live range;
3 if the operand's use begins a live range, attempting to allocate a register;
4 if the operand's use ends a live range, marking the register associated with the
5 operand as available for reallocation.

1 37: The system of claim 35, wherein the register allocator is capable of, when attempting
2 to allocate a preserved register:
3 determining if a preserved register is available for allocation from a list of

- 4 previously used preserved registers; and
- 5 if so, allocating the available preserved register for the live range defined by the operand .

- 1 38: The system of claim 37, wherein the register allocator is capable of, if a preserved
- 2 register is not available from the list of previously used preserved registers:
- 3 attempting to find a tentative live range that includes a preserved register that is in
- 4 the list of previously used preserved registers; and
- 5 if successful, allocating this preserved register for the live range defined by the operand,
- 6 and reserving a different preserved register for the tentative live range.

- 1 39: The system of claim 38, wherein the register allocator is capable of, if the attempt to
- 2 find a tentative live range that includes a preserved register that is in the list is not
- 3 successful:
- 4 adding a new preserved register to the list of previously used preserved registers; and
- 5 allocating this new preserved register to the live range defined by the operand.

- 1 40: The system of claim 35, wherein the register allocator is capable of, when attempting
- 2 to allocate a scratch register:
- 3 determining if a scratch register is available for allocation;
- 4 if so, allocating a scratch register for the live range defined by the operand;
- 5 if not,

6 attempting to steal a scratch register from a live range marked as tentative;
 7 determining if the attempted theft was successful;
 8 if so, allocating the stolen scratch register to the live range defined by the
 9 operand; and
 10 if not, attempting to allocate a preserved register.

1 41: The system of claim 35, wherein the register allocator is capable of, when treating a
 2 live range as tentative:
 3 if both a scratch register and a preserved register are available for allocation,
 4 reserving both the scratch register and the preserved register to the live range;
 5 allowing another instruction to be processed; and
 6 waiting for more information about the live range.

1 42: The system of claim 41, wherein the register allocator is capable of, when more
 2 information is available:
 3 if the last use of the live range has been scheduled, allocating the reserved scratch
 4 register to the live range defined by the operand, and freeing the reserved preserved
 5 register;
 6 if a function call has been scheduled before the last use of the live range or the
 7 reserved scratch register has been stolen, allocating the preserved register to the live
 8 range defined by the operand and freeing the reserved scratch register.

1 43: The system of claim 42, wherein the register allocator is capable of, when reserving a
2 preserved register for the tentative live range
3 preferring to reserve a currently unused preserved register that is in the list of previously
4 used preserved registers, before reserving a new unused preserved register for the
5 tentative live range .

1 44: The system of claim 43, wherein the register allocator is capable of, when treating a
2 live range as tentative:
3 if a scratch register is not available for allocation, attempting to allocate a
4 preserved register to the live range.

1 45. The system of claim 35, the register allocator is capable of:
2 if no proper registers are available for allocation, inserting a register spill; and
3 either re-attempting to allocate a register, or selecting another instruction to
4 schedule.

1 46. The system of claim 35, wherein the register allocator includes:
2 a scratch register allocator to allocate scratch registers to live ranges that do not
3 span a function call;

- 4 a preserved register allocator to allocate preserved registers to live ranges that
- 5 span a function call; and
- 6 a tentative register allocator to allocate either a scratch or a preserved register to
- 7 live ranges that are not immediately known whether or not they span a function call.